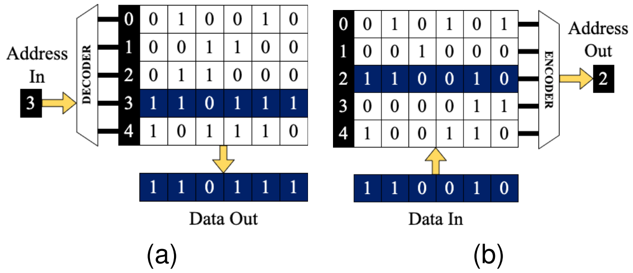
Advancements in Content-Addressable Memory (CAM) Circuits: State-of-the-Art, Applications, and Future Directions in the AI Domain

Tergel Molom-Ochir, Brady Taylor, *Graduate Student Member, IEEE*,
Hai Li, *Fellow, IEEE*, and Yiran Chen, *Fellow, IEEE*

***Abstract*— Content-Addressable Memory (CAM) circuits, distinguished by their ability to accelerate data retrieval through a direct content-matching function, are increasingly crucial in the era of AI and increasing data computation. With the rise of AI models, hardware matching and hashing capabilities become essential, underscoring the need for a comprehensive survey of this evolving technology. This survey explores various CAM types across circuit designs and technologies, highlighting contributions to fields such as Machine Learning and genomics. We review 37 CAM cell designs, focusing on emerging trends in area and energy efficiency, pivotal for next-generation computing. Furthermore, we discuss current challenges and suggest future research directions in CAM technology.**

***Index Terms*— Associative memory, Machine Learning, in-memory computing, semiconductor devices, digital circuits, analog circuits.**



0	0	1	0	0	1	0
1	0	0	1	1	0	0
2	0	1	1	0	0	0
3	1	1	0	1	1	1
4	1	0	1	1	1	0

Data Out					
1	1	0	1	1	1

0	0	1	0	1	0	1
1	0	0	1	0	0	0
2	1	1	0	0	1	0
3	0	0	0	0	1	1
4	1	0	0	1	1	0

Data In					
1	1	0	0	1	0

Fig. 1. Basic (a) RAM and (b) CAM operation.

I. Introduction

CONTENT-ADDRESSABLE Memory (CAM) is a specialized type of computer memory used in high-speed search applications [1]. Unlike traditional address-in, data-out memory types such as Random Access Memory (RAM), CAM memory functions with a data-in, address-out principle, allowing for rapid and efficient data searches [2], shown in Fig. 1. This parallel computing capability makes CAM highly relevant in applications such as networking [3], [4], genomics [5], [6], [7], [8], [9], [10], [11], [12], databases [13], [14], optical computing [15], [16], [17], [18], [19], real-time image processing [20], [21], and Machine Learning models [22], [23], [24], [25], [26], [27].

CAM operates by comparing input search data against a table of stored data. If data matching the input is found, the address of that data is returned [5]. There are several types of CAMs, with ternary CAM (TCAM) being the most common. TCAM stores data bits in three states: 0, 1, and X or "dont care", which returns a match regardless of the input [1], [28], [29], [30], [31], [32], [33], [34], [35], [36]. This flexibility allows TCAMs to handle more complex search queries efficiently. Analog CAM arrays store data as ranges of acceptable values, with analog inputs provided for matching. If an input value falls within the stored range of a cell, it is considered a match for that cell. This capability is useful for applications that require the matching of continuous or multilevel data values [37]. Lastly, differentiable CAMs handle all analog inputs, storage, and outputs, offering greater flexibility in search operations. Unlike standard analog CAMs, differentiable CAMs provide an analog output that indicates the degree of match between the input and the stored data [38]. This allows closest match search between analog input and stored values.

The associative or parallel search mechanism makes CAMs extremely fast at searching through large datasets, which is particularly beneficial for functions such as matching and hashing, where a CAM is used to search and decide whether a certain pattern exists in a large table of data. Binary and Ternary CAMs have been applied to data intensive genomics computations [5], [6], [7], [8], [9], [10], [11], [12], hamming distance calculations [39], and hashing [2], [15], [16], [18], [19], to name a few. Moreoever, Analog CAM's unique search mechanism were applied to Machine Learning model mapping and acceleration [37].

In big data and AI applications where vast amounts of data need to be accessed and processed rapidly, such as those found in computer vision and natural language processing, in-memory computing solutions are emerging to address the 'memory wall' challenge in AI hardware trends, where the speed of CPUs and GPUs is significantly obstructed by the latency and bandwidth limitations of traditional memory hierarchies [40]. CAM's ability to perform parallel searches in hardware significantly enhances the performance of systems

Received 30 August 2024; revised 27 November 2024; accepted 6 January 2025. This work was supported by the National Science Foundation (NSF) under Grant IIS-2332744, Grant ECCS-2328712, Grant CCF-2328805, and Grant CNS-2112562. This article was recommended by Associate Editor M. Lopez-Vallejo. *(Corresponding author: Tergel Molom-Ochir.)*

The authors are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27705 USA (e-mail: tergel.molom-ochir@duke.edu).

Digital Object Identifier 10.1109/TCSI.2025.3527309

that require fast data search and processing in-memory. In a CAM, the memory array is designed to broadcast input data to all rows of stored data and compare them simultaneously. Unlike sequentially checking each entry, CAM's parallel search can locate the matching entry in a single operation, achieving constant time complexity O(1) [22], [23], [24], [25]. This reduces latency and maximizes bandwidth utilization by minimizing the need for data to transfer back and forth between the memory and the processing units, overcoming the memory wall.

The realm of CAM technology is rapidly advancing, and a comprehensive survey is needed to understand the current advancements in the age of AI, and future directions. This survey aims to fill the gap by providing an in-depth analysis of the latest developments in CAM technologies, and applications in the AI landscape.

In this paper, we classify CAM technologies into several categories based on their operational mechanisms and underlying memory technologies. Section II will delve into CAM circuits, detailing different types of CAMs, as well as semiconductor and emerging non-volatile memory (NVM) technologies used for CAM implementation. Section III will provide a comprehensive analysis on emerging applications of CAM. Lastly, section IV discusses challenges, potential research directions, and anticipated advancements in CAM technology.

II. CAM Circuits

Unlike standard memories such as SRAMs, which are accessed by a specific memory address, CAMs allow data retrieval based on the content itself. In other words, CAM operates on search data-in and address-out, while standard memories operate on address-in and data-out principle.

The core cell, which compares a single bit of stored and query data, is the fundamental unit of any CAM array. CAM cells are connected vertically via bit lines, with a driver controlling the input, and horizontally via matchlines, where stored bit patterns reside. Each matchline has a sense amplifier to finalize the readout, and an encoder translates the sense amplifier outputs to the binary address of the matched matchline in case of a single best match. Most CAM arrays support exact matches, requiring all cells on a matchline to match. The basic structure of a CAM cell consists of two parts: storage circuitry and compare circuitry. The comparison operation in CAM cell involves two phases: precharge and evaluation. Despite numerous proposed and implemented core cell designs and reviews of existing designs, no comprehensive survey of recent CAM cell developments and emerging AI applications exists. This paper presents 39 different CAM core cells, shown in Table V, and discusses their applications in the age of AI.

In this section, we delve into the various circuits utilized in CAM technology. Different circuit designs provide various functionalities and performance characteristics, meeting diverse application needs. CAMs are classified by their underlying concepts, and the technologies used for their implementation. This section explores several types of CAMs, each with unique advantages and limitations, and technologies used for their construction.

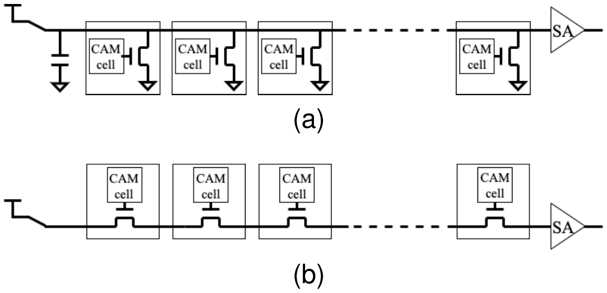


Fig. 2. Basic (a) NOR-type and (b) NAND-type CAM.

A. Matchline Architecture

CAM architectures can be broadly classified into two types: NOR and NAND, each with distinct characteristics as shown in Fig. 2.

In NOR-type architecture, each cell has a pull-down transistor connected to the match line. The match line is precharged, and during a search operation, the cell compares its stored value with the input value. If the values differ, the transistor discharges the match line via the pull-down transistor, indicating a mismatch. If all cells match, the pull-down transistor is not activated and the match line remains high. The NOR architecture enables simultaneous comparisons, making it fast. However, it consumes high power as it precharges and discharges multiple match lines with each search operation.

The NAND-type architecture operates using pass transistors. Each cell has a pass transistor connected to the match line, and the match signal propagates sequentially through these transistors. When a match occurs, the pass transistors are turned on, allowing the signal to pass to the sense amplifier. If there is a mismatch, the transistor remains off, blocking the signal. This architecture is power-efficient since it avoids precharging and discharging the match lines repeatedly. However, the sequential nature of the matching process makes it slower compared to the NOR architecture.

NOR-type is faster due to simultaneous comparisons. NAND-type is more power-efficient as it avoids continuous precharging and discharging. In summary, NOR-type architecture is suited for high-speed applications but at the cost of higher power consumption, whereas NAND-type architecture offers lower power usage but slower operation [36], [41], [42]. Balancing these characteristics is key to optimizing CAM performance for specific applications. Systems that combine the best of two matchline architectures has been developed with the goal to achieve high-speed, low-power CAM systems [28].

B. Architecture and Peripherals

Fig. 3 depicts the basic structure for a CAM system, necessary for the execution of high-speed search operations. Integral elements in this include pre-charge circuitry, precharging match-lines before each search; a query register that will store and broadcast the input search word to the searchlines; match line sense amplifiers (MLSAs) that will sense a match or a mismatch; and an encoder to convert the match results into a binary address. Depending on the application and architectural requirements, MLSAs can be designed to output

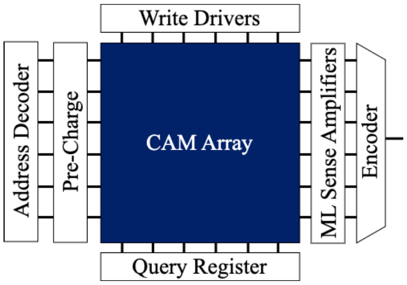


Fig. 3. CAM architecture with key peripherals for search and update operations.

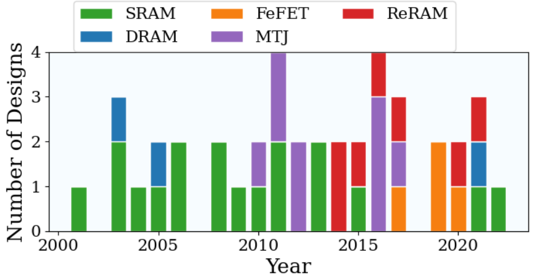


Fig. 4. Cell designs across different memory technologies over time.

exact match/mismatch or Hamming distances. In addition, address decoders and write drivers enable writing and updating stored data, hence the system is programmable. Other applications may require more peripherals to make better functionality. Examples of such are multiple match resolvers used in networking to handle simultaneous matches, priority encoders to determine the highest-priority match for routing application, and segmented matchlines to optimize power in energy-sensitive designs. In simpler read-only CAM systems, the write drivers and address decoders can be removed to further simplify the design. This modularity allows the architecture to be optimized for particular performance and power requirements.

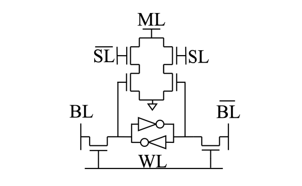
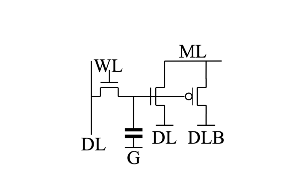
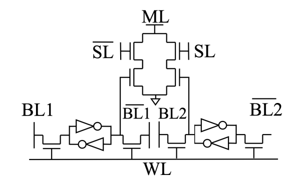
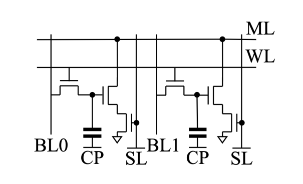
C. CAM Cell Concepts

Depending on the the specific needs and limitations of the application, different CAM circuits have been realized over the years Fig. 4. Predominantly implemented using SRAM technology, with digital CAMs, the inputs, storage, and output are represented as binary values (0 and 1), while digital ternary storage adds a “don’t care” state (X) for flexibility. Recently, analog CAM concept was introduced [37]; analog CAMs allow input and storage values ranging from 0.0 to 1.0, facilitating multilevel matching, with analog ternary storage also including the X state. Differentiable CAMs operate fully in analog, allowing unique output types through differentiable functions. Digital CAMs are discussed in depth in Section. II-C.1, Analog CAMs in Section. II-C.2, and Differentiable CAMs in Section. II-C.3. Various types of existing CAM concepts are shown in Table I.

TABLE I
Types of CAM Concepts

Input	Storage (B or T)	Output	Circuit
Digital	Digital	Digital	B/T CAM [31]
Analog	Analog	Digital	Analog CAM [37]
Analog	Analog	Analog	Differentiable CAM [38]

TABLE II
Cell designs for Binary and Ternary CAM Using SRAM (10T Binary, 16T Ternary) and DRAM (3T1C Binary, 6T2C Ternary) Technologies [31], [43], [44]

	SRAM	DRAM
Binary	 ML $\overline{\mathrm{SL}}$ SL BL $\overline{\mathrm{BL}}$ WL	
Ternary	 ML $\overline{\mathrm{SL}}$ SL BL1 $\overline{\mathrm{BL1}}$ BL2 $\overline{\mathrm{BL2}}$ WL	

1) Digital CAMs: CAMs are specialized devices designed for high-speed search operations. These devices are implemented using SRAM and DRAM technologies and function entirely with digital inputs, storage, and outputs. At the cell level, Binary CAM cells perform an XNOR operation between the stored bit and the search bit. If the stored and search bits are the same (either both 0 or both 1), the cell outputs a “1” (match); otherwise, it outputs a “0” (mismatch). On a row level, if all cells in a row match the input data, the row outputs a “1” on the matchline, and the corresponding memory address is returned. For instance, as shown in Fig. 1b, if the input data is 110010, the CAM will compare this input with all rows in parallel. If the third row matches the input data, the CAM outputs a “1” for this row and returns the address “2” via the encoder. Ternary CAMs (TCAMs) operate similarly to binary CAMs but include a third state, “don’t care” (X), which can match any input bit. For example, a TCAM row storing 10 × 1 can match input values 1011 and 1001. TCAMs provide straightforward match or mismatch signals. While digital CAMs are advantageous for their simple search capabilities, they often suffer from large physical sizes due to the complexity of their digital circuitry. Table II show example binary and ternary SRAM implmementation based on the 6T SRAM cell.

2) Analog CAMs: Analog CAMs operate with analog inputs and storage, enabling the processing and storage of a wide range of continuous data values [37]. Typically realized using emerging non-volatile memory (NVM) technologies, they are valuable for applications requiring precise data control. Despite using analog inputs and storage, the output is digital (match or mismatch). This hybrid approach can increase storage density and design efficiency, but may introduce noise-related challenges. This design is ideal for multilevel and non-binary state matching.

Analog CAMs store data as ranges between 0 and 1. When an analog input is provided, it is compared to these ranges. If the input falls within a stored range, it is considered

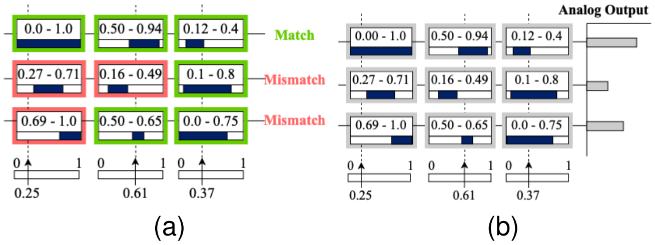


Fig. 5. Operation of (a) analog CAM with binary matching results and (b) differentiable CAM with analog matching outputs.

a "1" (match). For example, if a cell stores a range between 0.36 and 0.75 and the input is 0.60, the cell will output a match. If the input is outside this range, a "0" (mismatch) is returned. When all cells on a matchline output a match, the address of that matchline is returned.

3) Differentiable CAMs: Differentiable CAMs, which operate fully in analog, store data as ranges and receive analog inputs, providing an analog output that indicates the degree of match for each row rather than a simple "True" or "False" result as can be seen in Fig. 5. If a cell stores a range between 0.36 and 0.72, an input of 0.24, being closer to the stored range, will cause a smaller current in aMLlo and a slower discharge of the matchline, resulting in a higher analog output (weak mismatch) compared to an input of 0.12 (strong mismatch). For any value within the range, i.e. 0.48, the ML stays charged, indicating a match. This all-analog implementation links CAMs to analog crossbar arrays, enabling fast searches to determine the degree of match between tables of analog values and analog inputs, thus broadening the scope of applications that can perform similar searches in analog.

D. Semiconductor Technologies for CAM

1) SRAM: The storage circuitry is usually a SRAM cell, which is often implemented using six-transistors (6T) [29], [31], [34], [35], [41], [45], [46], [47]. The 6T cell comprises two cross-coupled inverters and two access transistors. Compare circuitry includes transistors that connect to the matchline and search lines. These transistors are responsible for comparing the stored bit with the search bit. Table II shows example binary and ternary SRAM cell designs.

This structure allows stable and low power data storage without the need for periodic refreshing, making it ideal for applications that require high-speed memory. This structure is challenged by scaling issues, resulting in higher production costs and larger cell sizes.

The two cross-coupled inverters create a stable storage element that holds a single bit of data. The remaining two nMOS transistors act as access transistors controlled by the word line. When writing data, the word line is activated, allowing the data bit to be written into the storage nodes through the bit lines. The data is stored at the intersection of the two cross-coupled inverters, providing a stable state as long as power is supplied. During a read, the word line is activated again, allowing the stored data to be read out through the bit lines.

A typical comparison goes as follows in a NOR-type CAM. The match line is precharged to a high voltage (logical 1). The search lines carry the search data. If any bit in the stored data does not match the search data, the corresponding transistor will pull the match line to a low voltage (logical 0). Thus, the match line remains high only if all bits match. When a mismatch is detected, the match line discharges through the nMOS transistors corresponding to the mismatched bits, pulling it low.

SRAM-CAMs are fast and have high endurance. However, they are volatile, have higher power consumption and suffer from large area. SRAM-CAMs are best for high-speed applications, such as networking routers and switches, and high-speed caches.

2) DRAM: Dynamic Random Access Memory (DRAM) based CAMs are characterized by their single transistor and capacitor configuration, which supports a compact and high-density design. This structure, while economical and capable of achieving higher cell density than SRAM-based CAMs, necessitates frequent refresh cycles due to charge leakage from the capacitors. This inherent volatility impacts the overall speed of memory access. Table II shows example binary and ternary DRAM cell designs.

Data in DRAM-based CAMs is stored as an electrical charge in a capacitor, indicative of binary data (1s and 0s). Each cell includes an access transistor, controlled by a word line, that regulates whether the capacitor charges (to store a '1') or discharges (to store a '0'). The need for periodic refresh cycles to replenish charge leakage is crucial for maintaining data integrity.

The comparison mechanism within DRAM-based CAM cells employs two key transistors that link the match line (ML) to ground. These transistors represent the stored bit and the inverse of the search bit. During a search operation, if the stored data matches the input query, only one transistor activates, preventing the ML from discharging. Conversely, a mismatch activates both transistors, discharging the ML through a direct path to ground. The state of the ML—either holding its precharged level in the event of a match or discharging in the case of a mismatch—is detected by match line sense amplifiers to confirm the presence or absence of a match. During a match, ideally, no current flows through the match line, and it remains at its precharged level. If there's a mismatch, the access transistor allows current to flow from the match line to ground, pulling the voltage down.

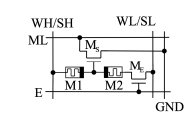
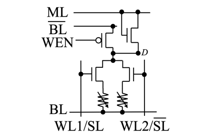
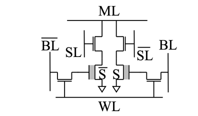
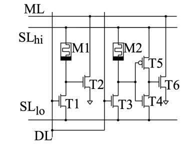
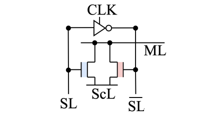
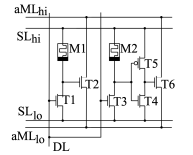
DRAM-based CAMs are ideal for large-scale memory applications where cost and density are prioritized over speed. Their use is particularly advantageous in fields requiring substantial memory resources, such as database acceleration, machine learning inference, and big data analysis. Despite their higher density and cost-effectiveness, the slower access times and the need for regular refreshes due to volatility should be considered when choosing memory solutions for high-speed applications.

E. Emerging Non-Volatile Memory (NVM) Technologies

1) Resistive RAM (RRAM/ReRAM): ReRAM-based CAM cells typically integrate transistors and memristors (ReRAM devices) to form a compact and efficient memory cell. A common structure includes configurations like 2T2R

TABLE III

Cell Designs for CAM Using Emerging Non-Volatile Technologies: Ternary ReRAM (2T2M) [48], Ternary MTJ (4T-2MTJ) [49], Ternary FeFET (4T-2FeFET) [50], Analog ReRAM (6T2M) [37], and Differentiable (6T2M) [38]

	ReRAM	MTJ	FeFET
Ternary			
Analog	 ML SLhi M1 M2 T5 T2 T6 T1 T3 T4 SLlo DL		
Differentiable	 aMLhi SLhi M1 M2 T5 T6 T2 T1 T3 T4 SLlo aMLlo DL		

(two transistors and two ReRAMs) or 3T1R (three transistors and one ReRAM) for ternary CAMs and 6T2M (six transistors and two memristors) for Analog and differentiable CAMs. The transistors are used for access control and signal amplification, while the memristors are employed for data storage due to their resistive switching capabilities. For example, in a 2T2R cell design: Two transistors (T1 and T2) are used to control the read and write access to the cell. Two ReRAM devices (R1 and R2) are used to store the binary data, leveraging their high resistance (HRS) and low resistance (LRS) states to represent logic '1' and '0'.

Data in ReRAM-based CAM cells is stored in the resistive states of the memristors. Each memristor can switch between a high resistance state (HRS) and a low resistance state (LRS), representing binary data. During the write operation, a voltage is applied across the ReRAM device to change its resistance state. For example, applying a higher voltage might set the device to LRS (logic '0'), while applying a lower voltage or reverse polarity might set it to HRS (logic '1'). The resistance state of the ReRAM is non-volatile, meaning it retains its state even when the power is turned off, thus storing the data persistently. ReRAM cells are smaller, allowing for higher density memory arrays. They are non-volatile; ReRAM retains data without power, unlike SRAM and DRAM which require constant power. Moreover, ReRAM-based designs generally consume less power, especially in idle and search operations.

For digital binary or ternary CAMs, in a 2T2R cell shown in Table III, the current through the cell during a read or search operation is determined by the combined resistance of R1 and R2. For a match (both memristors in the same state), the current flow will be as expected (either high if both are LRS or low if both are HRS). For a mismatch (memristors in different states), the current will be different from the expected value (one high and one low resistance, resulting in an intermediate current).

In Analog CAMs, implemented as 6T2M [37] shown in Table III, which checks a range of values, each side of the cell evaluates a 'greater than' or 'less than' condition. When an input voltage is applied to the Data Line (DL), the conductance through transistors T1 and T3 is compared to the stored conductances on memristors M1 and M2, respectively. If the conductance through T1 is greater than that of M1, transistor T2 remains off, preventing the Match Line (ML) from being pulled down, indicating DL > M1. Conversely, if DL < M1, there is a conductive path from the Source Line high (SLhi) to T2, turning T2 on and pulling ML down to ground, indicating a mismatch.

The circuit has been modified to operate as a differentiable CAM (dCAM) by adding the ability to sense the discharge current on aMLlo, representing a distance metric between the input and stored data. In the former case, the current changes as aMLhi discharges but is generally assumed constant while T2 and T6 are in saturation. In the latter case, the current is constant and depends on the voltage on aDL and the stored conductance. If the input is close to the stored values, a small current flows in aMLlo, whereas a significant difference results in an increased current, rapidly discharging aMLhi. An example cell design is shown in Table III.

Further, Khan and Rashid [48] discusses a hybrid ternary CAM using memristors to minimize area and wiring complexity. Bazzi et al. [51] introduced new analog CAM cell designs using memristors with an emphasis on the gain of cell parts. This combination of memristors and transistors introduces a more compact, power-efficient, reliable, and high-performance memory architectures, which is highly sought after in big data and IoT.

ReRAM-CAMs are non-volatile, high density, and low power but are limited in terms of endurance, and has variability in resistance states. They are best suited for energy-efficient applications such as machine learning models acceleration, and edge computing.

2) Magnetoresistive RAM (MRAM): The MTJ based non-volatile ternary content-addressable memory (NV-TCAM) cell consists of transistors and magnetic tunnel junctions (MTJs). Table III shows an example of a 2MTJ-4T design. The MTJs act as resistors with two possible states based on their magnetization: parallel alignment (low resistance, RL) and antiparallel alignment (high resistance, RH). The transistors include NMOS transistors for connecting the MTJs to the search lines (SL and $\overline{\text{SL}}$), and a PMOS transistor functioning as a voltage keeper to stabilize the match line (ML) voltage during the evaluation phase. Additionally, an NMOS transistor connected to the ML acts as a diode to control the discharge path.

Data in the NV-TCAM cell is stored in the MTJs based on their resistance states. For a binary '0', R1 is set to RH and R2 to RL. For a binary '1', R1 is set to RL and R2 to RH. If the data is 'don't care' (X), both R1 and R2 are set to RH. The resistance state of each MTJ is determined by the alignment of the magnetizations in its two ferromagnetic layers, which can be altered by applying a specific current through the MTJ.

During the precharge phase, SL and $\overline{\text{SL}}$ are grounded, and the ML voltage (VML) is precharged to VDD using an external precharge transistor. In the evaluation phase, SL and $\overline{\text{SL}}$ are set to opposite voltages depending on the search data (VDD and GND or vice versa). If the stored data matches the search data, the connected MTJ remains in the high resistance state (RH), resulting in a high D-node voltage ($V_{@D}H$). VML discharges from VDD to $V_{@D}H + V_{TH}Keeper$ (the threshold voltage of the voltage keeper), cutting off the voltage keeper. If there is a mismatch, the connected MTJ is in the low resistance state (RL), resulting in a low D-node voltage ($V_{@D}L$). VML discharges to $V_{@D}L + V_{TH}Keeper$. The match line (ML) sense amplifier detects the voltage difference (ΔVML) to determine if the TCAM word matches the search data.

MTJ-CAMs are non-volatile, high-speed, and have high endurance. These advantages come at a high cost and fabrication complexity. They are best suited for high-speed non-volatile memory applications as they combine speed and non-volatility.

3) Ferroelectric Field-Effect Transistor (FeFET): A ferroelectric field-effect transistor (FeFET) incorporates a ferroelectric material into its gate dielectric. This material exhibits unique properties, allowing it to maintain a polarization state even without a power supply. A ternary FeFET CAM cell typically consists of two FeFETs and four transistors, shown in Table III.

Data storage in FeFET-based CAM cells is achieved through the polarization states of the ferroelectric material. When a voltage is applied to the gate, it polarizes the ferroelectric layer, writing binary data (0 or 1) based on the direction of the polarization. This polarization remains stable even when the power is turned off, ensuring non-volatile data storage. In a 2FeFET TCAM, each cell uses two parallel FeFETs connected to a matchline (ML) and sourceline (ScL). During a search operation, a specific voltage is applied to the gate of each FeFET in the CAM array. The current response of the FeFET indicates whether the stored polarization state matches the input data. If the stored state matches the input data, the current does not flow from ML to GND, signifying a match.

A logic ‘1’ is written by applying V_write to the gate (BL/SL) and GND to the source (ScL), while a logic ‘0’ is written by reversing these voltages. The don’t care state (X) is stored by writing logic ‘0’ into both FeFETs, allowing both transistors to hold ‘0’. During a search operation, the matchline (ML) is precharged high, and search voltages (V_search) are applied to the gates (SL/$\overline{\text{SL}}$) according to the input data—V_search for logic ‘1’ and 0 for logic ‘0’. The inputs to the transistors ($\overline{\text{SL}}$ and SL) and the stored states (S and $\overline{S}$) determine whether the pull-down paths are ON or OFF. If there is a match, both pull-down paths remain OFF, keeping the ML high. In the event of a mismatch, at least one pull-down path is ON, discharging the ML. This design ensures efficient comparison operations with the current flow indicating the match or mismatch, while the don’t care state keeps the ML high regardless of the input [50], [65].

Leveraging the multilevel-cell states in FeFETs, a 2FeFET-based CAM design, shown in III, can store continuous analog values by setting upper and lower bounds using two FeFETs connected to an inverted searchline (SL). Each FeFET defines the bounds for matching the input voltage (V_{SL}). During a search, the ML is precharged, and if V_{SL} falls within the stored range (between the upper and lower bounds set by the FeFETs’ threshold voltages), the ML remains high, indicating a match. If V_{SL} is outside this range, one of the FeFETs turns on and discharges the ML, indicating no match. This configuration allows for flexible and efficient range-based searching and matching, suitable for applications requiring continuous range storage and multi-bit quantized searches [66].

TABLE IV

COMPARISON OF CAM TECHNOLOGIES

Feature	SRAM CAM	DRAM CAM	NVM CAM
Speed	High	Moderate	High (MRAM)
Density	Low	High	Very High
Power Efficiency	Low	Moderate	High
Non-volatility	No	No	Yes
Cost	High	Low	High (MRAM), Low (ReRAM)
Endurance	High	Moderate	High (MRAM), Limited (ReRAM)
Applications	Networking, caches	Databases, ML	AI, Genomics, IoT

Their fast switching capabilities make them ideal for high-speed search operations, while their non-volatile nature ensures data retention without power. However, they have limited endurance due to the wear on the ferroelectric material and involve complex materials and fabrication processes. FeFET-CAMs are well-suited for low-power and high-speed applications, such as genomic data processing and real-time data analytics, in-memory computing, and memory-augmented neural networks [50], where FeFET-based TCAMs can drastically reduce energy use and latency.

F. Comparison of CAM Technologies

As shown in Table IV, comparison among SRAM, DRAM, and NVM CAM designs shows each technology has different strengths and trade-offs. SRAM-based CAMs are highest in terms of speed and endurance and thus seem well suited for high-end applications, including networking; however, they have very high power consumption and area footprint and, therefore, are not scalable. On the other hand, DRAM-based CAMs have a high density and low cost, enabling compact solutions for large memory-hungry applications like machine learning inference but are inherently volatile, with potentially lower access times; otherwise, NVM CAMs, including ReRAM, MRAM, and FeFET-based architectures, are characterized by non-volatility, energy efficiency, and compact form factor. Each NVM type has its unique strengths: ReRAM has high density and low power but faces endurance challenges; MRAM is high-speed and offers good endurance at a higher cost; FeFET allows fast multi-state storage but suffers from fabrication complexity and poor endurance. The final choice of CAM technology will depend on the specific application requirements of speed, power efficiency, cost, and memory density.

III. EMERGING APPLICATIONS

In the space of memory technology, the fast-growing development and transformation of CAM has been characterized

TABLE V
CAM CELL DESIGNS

Design	Technology	Process Node (nm)	Area (μm^2)	Search Energy (fJ/bit/search)	Year
16T NOR [1]	SRAM	130	6.73	-	2003
16T NAND [28]	SRAM	100	22.40	0.70	2005
10T [29]	SRAM	180	11.15	-	2001
16T NOR [30]	SRAM	65	1.69	1.98	2013
6T [31]	SRAM	28	0.15	0.60	2015
10T [47]	SRAM	65	3.30	0.77	2013
9T [52]	SRAM	130	20.21	1.87	2010
13T [41]	SRAM	180	30.00	2.82	2011
12T NAND [32]	SRAM	130	12.93	-	2008
8T [46]	SRAM	28	0.74	-	2022
4T [33]	SRAM	180	17.54	-	2003
16T NOR [34]	SRAM	130	7.10	-	2004
12T NAND [35]	SRAM	130	15.22	0.44	2009
12T [36]	SRAM	180	19.47	1.42	2008
10T [45]	SRAM	28	2.66	1.02	2021
16T NAND [42]	SRAM	65	7.05	0.17	2011
6T2C [43]	DRAM	130	3.59	-	2005
8T4C [53]	DRAM	130	4.79	-	2003
2T1C [44]	DRAM	28	0.10	-	2021
3T1R [54]	ReRAM	90	1.57	0.51	2015
2T2R [55]	ReRAM	90	0.41	-	2014
2.5T1R [56]	ReRAM	65	0.59	0.28	2016
4T2R [57]	ReRAM	180	9.70	-	2014
6T2M [37]	ReRAM	16	0.52	-	2020
3T1R [58]	ReRAM	90	0.87	0.51	2017
2T2M [48]	ReRAM	65	6.11	0.87	2021
10T4MTJ [59]	MTJ	45	2.78	40.50	2017
4T2MTJ [49]	MTJ	90	3.14	-	2012
1T1MTJ [60]	MTJ	140	1.25	-	2011
9T2MTJ [61]	MTJ	90	10.35	0.73	2012
11T3MTJ [62]	MTJ	180	36.00	7.10	2010
6T2MTJ [63]	MTJ	90	10.35	1.04	2011
3T2MTJ [64]	MTJ	45	0.13	-	2016
2FeFET [50]	FeFET	45	0.15	-	2019
4T2FeFET [65]	FeFET	45	0.65	-	2017
2FeFET [66]	FeFET	45	0.05	0.18	2020
2FeFET [67]	FeFET	45	0.15	0.40	2019

by significant milestones. In the early 1990s, the application of CAM started with associative memory [76] and processing [77] and image processing [20], [21]. With the recent trend in processor-memory gap and the rise of the AI models, in-memory computing has become a promising direction and CAMs have shown potential solutions in overcoming the "memory wall." From 2020 onwards, CAMs developments were powered by cutting-edge emerging developments such as ferroelectric devices and compute-in-memory arrays, exploiting CAM's core fundamental operational principles for AI hardware and data processing at a large scale [5], [22], [23], [24], [25], [26], [27], [37], [50], [65], [66], [67]. Fig. 8 shows the CAM application trends over the last ten years.

A. Machine Learning

Recent advancements in tree-based machine learning models and analog CAMs have demonstrated significant potential for in-memory acceleration. Notably, memristive devices have been utilized to build analog CAMs that accelerate these models [24], and further developments have enabled the acceleration of Deep Random Forests on CAMs [75]. In the realm of neural networks, redundant analog-to-digital conversions in RRAM-based CNN accelerators were addressed by BRAHMS, a hybrid analog RAM and CAM system that enhances performance and energy efficiency [23]. Efficient NN acceleration on GPGPU was achieved by storing important features on CAM [71], while another study introduced a CAM-based binarized neural network accelerator using time-domain signal processing [72]. Additionally, Ferroelectric ternary CAM was used for one-shot learning via Memory Augmented Neural Network [67]. In transformer networks, CAM-based process-in-memory techniques have been integrated with novel attention mechanisms to overcome computational and memory bandwidth bottlenecks. iMCAT, an architecture combining crossbars and CAMs for Transformer network acceleration, utilized locality-sensitive hashing to filter sequence elements by importance [26]. Furthermore, iMTransformer [27] and RACE-IT, a Reconfigurable Analog CAM-crossbar Engine, have been proposed to accelerate in-memory Transformer operations, with RACE-IT enabling efficient analog execution of various non-MVM operations within Transformer models [25].

TABLE VI
PERFORMANCE IMPROVEMENTS OVER SOTA ACROSS CAM-BASED APPLICATIONS

Year	Application	Improvement
2014	IP Lookup [68]	2x smaller index TCAM, O(logM) search
2015	Forwarding Engine [69]	1.67x energy, 1.43x cost efficiency, 3 memory accesses per lookup
2016	Approximate Computing [70]	1.35x energy, 2.9x vs offline profiling
2017	NN Computation Optimization [71]	1.68x energy, 1.4x speed
2018	Binarized NN [72]	1.385x energy, 1.094x area, 2.4x speed
2019	Memory Augmented NN [67]	60x energy, 2,700x latency
2020	Finite State Machines [73]	25x throughput/watt
2021	Boolean Satisfiability Acceleration [74]	62-185x speed, supports 32M clauses
2021	Tree-Based ML Model [24]	1,000x throughput, 12x energy
2022	DNA Classification [6]	2.2x sensitivity, 1,200x throughput
2023	Transformer Acceleration [25]	10.7x speed, 1,193x energy
2024	Deep Random Forest [75]	106x energy (vs CPU), 10x (vs ReRAM)

B. Genomics

CAM's unique matching capabilities have significantly advanced genomic data processing, enhancing speed and efficiency. In 2020, PARC, a Processing-in-Memory architecture utilizing ReRAM-based CAM, was introduced to target the computationally intensive chaining step in DNA alignment [7]. This step, which involves ordering and aligning sequences based on similarity, is computationally demanding due to the large amounts of genomic data involved. In 2022, BioSEAL further advanced CAM applications in genomics, aiming to accelerate biological sequence alignment broadly [10]. In 2023, DASH-CAM, a dynamic storage-based CAM system for pathogen classification, highlighted the dynamic storage capabilities of CAM [5]. Additionally, ASMCap, employing capacitive multi-level CAM for approximate string matching in genomic sequence analysis, explored the potentials of non-ReRAM based CAMs [8]. These developments contribute to the unique applications of CAM technologies in data processing-intensive biological research and medical diagnostics.

C. Hashing and Similarity Searches

CAM's capabilities for direct data comparison and retrieval within the memory hardware itself makes it feasible to do similarity search calculations. Hamming distance calculations were performed in [16] and [18], streamlining the process of searching and matching patterns within the memory. Nearest neighbor searches were performed in-memory in [2] and [15] using TCAM and FeFET-based multi-bit CAMs, respectively. Moreover, allowing efficient processing of high-dimensional data, hashing is performed on the chip using CAMs in [19].

D. Specialized CAM Technologies

1) Optical CAM: Optical Content-Addressable Memory (OpticalCAM) enhances traditional CAMs with advanced photonic circuits, using light for search operations across memory entries. Compared to electronic CAMs, Optical-CAMs are significantly faster and more energy-efficient. In OpticalCAMs, search data encoded as light signals interact with stored optical data within the cell's memory structure. Match detection is performed using XOR functions implemented with semiconductor optical amplifiers (SOAs) and Mach-Zehnder Interferometers (MZIs), which determine if the search data matches the stored data. Data writing involves changing the optical state of the storage mechanism using SOA-MZI flip-flops, while reading data involves sending a probe light and measuring the output with photodetectors. Developments in optical CAM and RAM systems have achieved error-free 10 Gb/s operations using SOA-MZI-based optical flip-flops [15]. Additionally, address bit levels were increased to 2-bit, and all-optical CAM systems were further developed [17], [19]. Recently, ternary CAMs using optical multiplexing techniques have achieved speeds up to 10 Gb/s [18].

2) Quantum CAM: Quantum-dot Cellular Automata (QCA) utilizes electron positioning within quantum dots to represent binary information, offering a high-speed, low-power alternative to traditional CMOS technology. QCA-based CAM cells are highly efficient for nanoscale data storage and retrieval, with data stored by the spatial configuration of electrons in a cell, where binary states are determined by electron positions [78]. The search operation involves initializing the cells during a precharge period, followed by comparing the input data to the stored data along a matchline using QCA gates like the majority and minority gates. These gates check if the input electron's position aligns with the stored configuration, signaling a match if they do, otherwise, no signal is sent [79]. The architecture includes arrays of QCA cells and gates for individual addressing and comparison. Notable achievements of QCA technology include operational speeds in the nanosecond range and an area throughput of 0.14 μm^2 per cell [79].

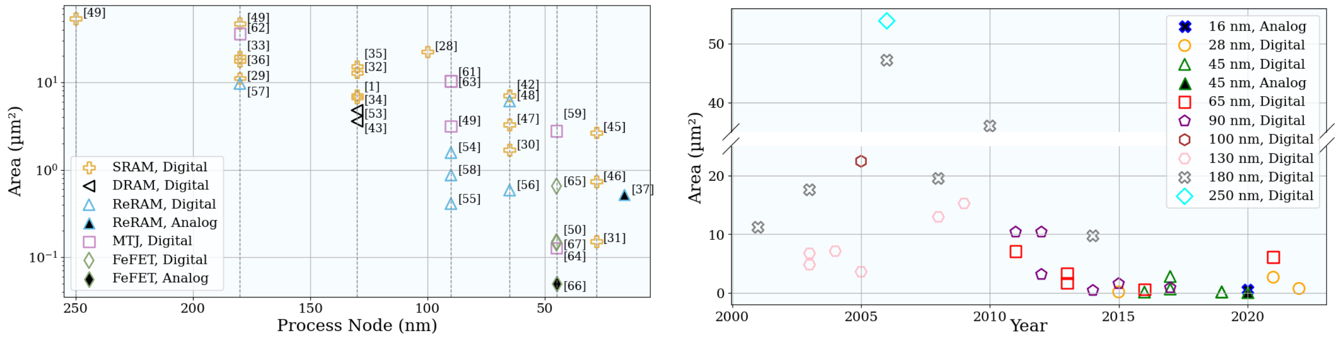
IV. CHALLENGES AND OUTLOOK

A. Challenges

CAMs face reliability challenges due to susceptibility to errors, affecting the accuracy and efficiency of CAM operations. These errors can cause incorrect data retrieval and increased latency, compromising the performance of systems that depend on data access and processing.

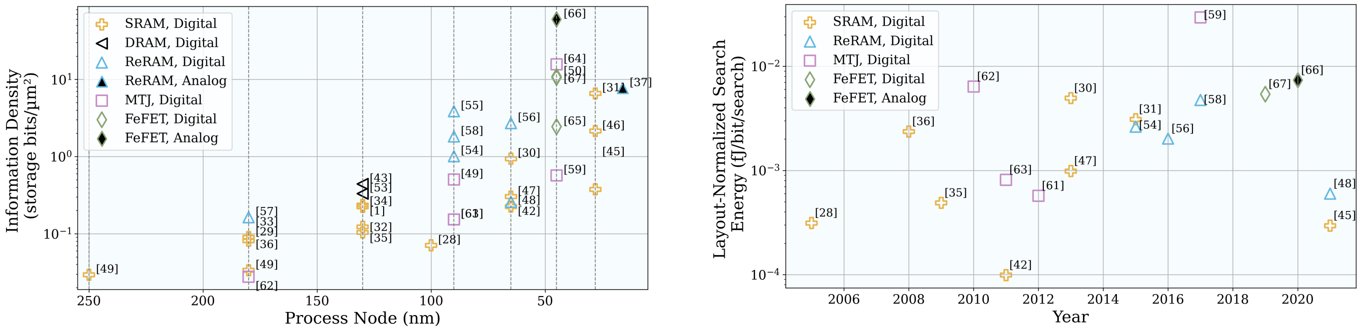
The challenge of maintaining accuracy in CAMs is further exacerbated by the continuous down scaling of technology nodes, making them vulnerable to soft errors caused by external electromagnetic radiation and internal voltage fluctuations and noise [80]. With an exact search function, where all cells on a row have to output a match to yield a row match, as the number of cells in a row increases, the probability of encountering an error rises, necessitating an error detection and correction mechanisms.

To address this challenge, researchers have been developing various error detection and correction schemes. Pontarelli et al. [81] proposed an error correction method based on the CAM/RAM system that does not alter the



(a) (b)

Fig. 6. CAM cell area analysis: (a) area scaling with process technology node and (b) published cell area trends from 2000 to 2023.



(a) (b)

Fig. 7. (a) Information density (bits stored per unit area) of CAM cells normalized by storage capacity across various process nodes. The trend highlights improvements in storage efficiency with technology scaling and different CAM implementations. (b) Layout-normalized search energy (Search Energy/k) versus publication year for analog and digital CAM designs implemented in FeFET, MTJ, ReRAM, SRAM, and DRAM technologies. Here, $k = \frac{\text{Area}}{(\text{Technology Node})^2}$ is used to normalize layout metrics for fair comparison. The relationship between area (μm^2) and search energy (fJ/bit/search) is captured by an exponential fit (Search Energy $= 0.621e^{0.0602\cdot\text{Area}}$) with an $R^2 = 0.749$, indicating a strong correlation.

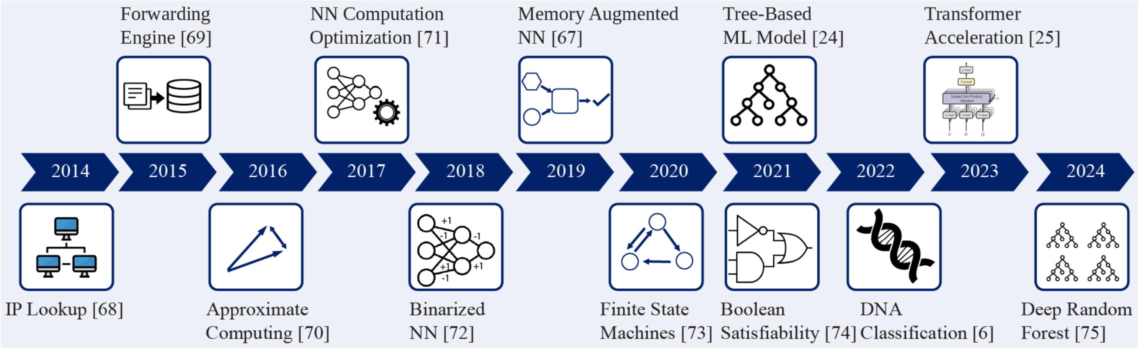


Fig. 8. Trends in CAM applications over time, highlighting key developments from 2014 to 2024. [6], [24], [25], [67], [68], [69], [70], [71], [72], [73], [74], [75].

CAM's internal structure. Varada and Agrawal [82] introduced a power-efficient TCAM architecture where the traditional priority encoder is replaced with multiplexers and a 2D parity technique is used for multi-bit detection and correction. Moreover, although Analog CAMs enable powerful capabilities such as acceleration of machine learning tasks and nonlinear activation functions, they also come with issues of error and reliability as they depend on memristors. Roth et al. [83] developed a technique to overcome the reliability issues by introducing coding schemes with minor additions to the hardware. These advancements highlight the ongoing efforts to overcome the reliability challenges in CAMs.

B. Directions

From the current trends in CAM technologies, the following areas are worth particular emphasis for future research:

(1) As depicted by the trajectory of CAM cell area reduction (Fig. 6) and information density increase (Fig. 7), we anticipate this trend to continue as technology nodes and cell designs advance every few years. Notably, the adoption of memristor devices is expected to grow

due to designer preference for lower area density, non-volatility, and analog capabilities.

(2) In the era of AI, CAM technologies are undergoing a significant transformation. Traditionally used as look-up tables, CAMs are now demonstrating how their matching capabilities can be used for Machine Learning tasks. Different types of CAM concepts, as illustrated in Table I, with varied search functions, will enable in-memory acceleration of operations within next-generation AI models. This highlights further exploration of CAM designs tailored especially for AI workloads. Table IV shows that the most beneficial application of AI acceleration is in the key areas of natural language processing, computer vision, and bioinformatics.

(3) Prior works have shown that the parallel processing capabilities of CAMs have high potentials to accelerate increasingly data and computations-intensive genomics sequencing on hardware. The exploration of CAMs for genomics applications is anticipated to grow with the expanding computational biology market, driven by the need for efficient and high-speed data processing. To this end, the extensions of different types of CAMs for genomic processing purposes and applications might be a meaningful future direction.

(4) Development of different CAM concepts, each with distinct input, storage, and output types, will enable hardware implementataton of diverse search functions through circuit-level innovation. Catering to a wide range of applications, this approach will enable more flexible search capabilities, such as best match, threshold match, and partial match. Circuit level innovation of various search functions will be an significant topic for future study.

V. Conclusion

This survey presents CAM circuits as a transformative technology in the semiconductor memory landscape. We reviewed various types of CAMs, including digital, analog, and differentiable CAMs, as well as their underlying technologies such as SRAM, DRAM, ReRAM, MRAM, and FeFET. CAMs have demonstrated their potential in enhancing and accelerating traditionally computationally expensive tasks such as machine learning algorithms, genomics data analysis and hashing. Future research should focus on developing various CAM concepts and search functions, efficient error correction schemes, integrating CAMs with emerging AI models, and exploring new applications in computational biology. The ongoing advancements in CAM technology are poised to address the computational demands of AI and computation intensive workloads, representing a significant leap toward faster and more efficient hardware-based computational methods.

Acknowledgment

The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, expressed or implied by the NSF.

References

[1] G. Kasai, Y. Takarabe, K. Furumi, and M. Yoneda, “200 MHz/200 MSPS 3.2 W at 1.5 V Vdd, 9.4 Mbits ternary CAM with new charge injection match detect circuits and bank selection scheme,” in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2003, pp. 387–390.

[2] A. Bremler-Barr, Y. Harchol, D. Hay, and Y. Hel-Or, “Ultra-fast similarity search using ternary content addressable memory,” in *Proc. 11th Int. Workshop Data Manage. New Hardw.*, May 2015, pp. 1–10.

[3] A. Ooka, S. Atat, K. Inoue, and M. Murata, “Design of a high-speed content-centric-networking router using content addressable memory,” in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 458–463.

[4] P. James-Roxby and D. Downs, “An efficient content-addressable memory implementation using dynamic routing,” in *Proc. 9th Annu. IEEE Symp. Field-Program. Custom Comput. Mach. (FCCM)*, Apr. 2001, pp. 81–90.

[5] Z. Jahshan, I. Merlin, E. GarzÓN, and L. Yavits, “DASH-CAM: Dynamic approximate search content addressable memory for genome classification,” in *Proc. 56th Annu. IEEE/ACM Int. Symp. Microarchitecture*. New York, NY, USA: Association for Computing Machinery, Oct. 2023, pp. 1453–1465, doi: 10.1145/3613424.3614262.

[6] E. Garzón et al., “Hamming distance tolerant content-addressable memory (HD-CAM) for DNA classification,” *IEEE Access*, vol. 10, pp. 28080–28093, 2022.

[7] F. Chen, L. Song, H. Li, and Y. Chen, “PARC: A processing-in-CAM architecture for genomic long read pairwise alignment using ReRAM,” in *Proc. 25th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2020, pp. 175–180.

[8] H. Zhong et al., “ASMCap: An approximate string matching accelerator for genome sequence analysis based on capacitive content addressable memory,” in *Proc. 60th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2023, pp. 1–6.

[9] L. Yavits, “DRAMA: Commodity DRAM based content addressable memory,” *IEEE Comput. Archit. Lett.*, vol. 23, no. 1, pp. 65–68, Jan. 2024.

[10] R. Kaplan, L. Yavits, and R. Ginosasr, “BioSEAL: In-memory biological sequence alignment accelerator for large-scale genomic data,” in *Proc. 13th ACM Int. Syst. Storage Conf.* New York, NY, USA: Association for Computing Machinery, 2020, pp. 36–48, doi: 10.1145/3383669.3398279.

[11] P. K. Lala, “A CAM (content addressable memory) architecture for codon matching in DNA sequences,” *Current J. Appl. Sci. Technol.*, vol. 10, no. 5, pp. 1–8, Jul. 2015. [Online]. Available: https://journalcjast.com/index.php/CJAST/article/view/197

[12] I. Merlin, E. Garzón, A. Fish, and L. Yavits, “DIPER: Detection and identification of pathogens using edit distance-tolerant resistive CAM,” *IEEE Trans. Comput.*, vol. 73, no. 10, pp. 2463–2473, Oct. 2024.

[13] H. Li, H. Jin, L. Zheng, and X. Liao, “ReSQM: Accelerating database operations using ReRAM-based content addressable memory,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 4030–4041, Nov. 2020.

[14] N. Bandi, S. Schnieder, D. Agrawal, and A. Abbadi, “Hardware acceleration for database systems using content addressable memories,” in *Proc. 1st Int. Workshop Data Manage. New Hardware (DaMoN)*, Baltimore, MD, USA. New York, NY, USA: Association for Computing Machinery (ACM), 2005, Paper 1–es, doi: 10.1145/1114252.1114262.

[15] A. Kazemi et al., “FeFET multi-bit content-addressable memories for in-memory nearest neighbor search,” *IEEE Trans. Comput.*, vol. 71, no. 10, pp. 2565–2576, Oct. 2022.

[16] L. Liu et al., “A reconfigurable FeFET content addressable memory for multi-state Hamming distance,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 6, pp. 2356–2369, Jun. 2023.

[17] G. Mourgias-Alexandris, C. Vagionas, A. Tsakyridis, P. Maniotis, and N. Pleros, “Optical content addressable memory matchline for 2-bit address look-up at 10 Gb/s,” *IEEE Photon. Technol. Lett.*, vol. 30, no. 9, pp. 809–812, May 1, 2018.

[18] J. S. Lee, J. Yoon, and W. Y. Choi, “In-memory nearest neighbor search with nanoelectromechanical ternary content-addressable memory,” *IEEE Electron Device Lett.*, vol. 43, no. 1, pp. 154–157, Jan. 2022.

[19] R. Mao et al., “Experimentally validated memristive memory augmented neural network with efficient hashing and similarity search,” *Nature Commun.*, vol. 13, no. 1, p. 6284, Oct. 2022, doi: 10.1038/s41467-022-33629-7.

[20] Y. C. Shin, R. Sridhar, V. Demjanenko, P. W. Palumbo, and S. N. Srihari, “A special-purpose content addressable memory chip for real-time image processing,” *IEEE J. Solid-State Circuits*, vol. 27, no. 5, pp. 737–744, May 1992.

[21] S. Panchanathan and M. Goldberg, "A content-addressable memory architecture for image coding using vector quantization," *IEEE Trans. Signal Process.*, vol. 39, no. 9, pp. 2066–2078, Aug. 1991.

[22] H. Zhu et al., "Fuse and mix: MACAM-enabled analog activation for energy-efficient neural acceleration," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Oct. 2022, pp. 1–9.

[23] T. Song, X. Chen, X. Zhang, and Y. Han, "BRAHMS: Beyond conventional RRAM-based neural network accelerators using hybrid analog memory system," in *Proc. 58th ACM/IEEE Design Autom. Conf. (DAC)*, Dec. 2021, pp. 1033–1038.

[24] G. Pedretti et al., "Tree-based machine learning performed in-memory with memristive analog CAM," *Nature Commun.*, vol. 12, no. 1, p. 5806, Oct. 2021, doi: 10.1038/s41467-021-25873-0.

[25] L. Zhao et al., "RACE-IT: A reconfigurable analog CAM-crossbar engine for in-memory transformer acceleration," 2023, *arXiv:2312.06532*.

[26] A. F. Laguna, A. Kazemi, M. Niemier, and X. S. Hu, "In-memory computing based accelerator for transformer networks for long sequences," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Feb. 2021, pp. 1839–1844.

[27] A. F. Laguna, M. M. Sharifi, A. Kazemi, X. Yin, M. Niemier, and X. S. Hu, "Hardware-software co-design of an in-memory transformer network accelerator," *Frontiers Electron.*, vol. 3, Apr. 2022, Art. no. 847069. [Online]. Available: https://www.frontiersin.org/articles/10.3389/felec.2022.847069

[28] S. Choi, K. Sohn, and H.-J. Yoo, "A 0.7-fJ/bit/search 2.2-ns search time hybrid-type TCAM architecture," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 254–260, Jan. 2005.

[29] P.-F. Lin and J. B. Kuo, "A 1-V 128-kb four-way set-associative CMOS cache memory using wordline-oriented tag-compare (WLOTC) structure with the content-addressable-memory (CAM) 10-transistor tag cell," *IEEE J. Solid-State Circuits*, vol. 36, no. 4, pp. 666–675, Apr. 2001.

[30] I. Hayashi et al., "A 250-MHz 18-Mb full ternary CAM with low-voltage matchline sensing scheme in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2671–2680, Nov. 2013.

[31] S. Jeloka, N. Akesh, D. Sylvester, and D. Blaauw, "A configurable TCAM/BCAM/SRAM using 28 nm push-rule 6T bit cell," in *Proc. Symp. VLSI Circuits (VLSI Circuits)*, Jun. 2015, pp. C272–C273.

[32] M. Sultan, M. Siddiqui, S. Arora, and G. S. Visweswaran, "A low-power ternary content addressable memory (TCAM) with segmented and non-segmented matchlines," in *Proc. IEEE Region 10 Conf.*, Nov. 2008, pp. 1–5.

[33] A. Igor, C. Trevis, and A. Sheikholeslami, "A ternary content-addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme," *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 155–158, Jan. 2003.

[34] A. Roth, D. Foss, R. McKenzie, and D. Perry, "Advanced ternary CAM circuits on 0.13 μm logic process technology," in *Proc. IEEE Custom Integr. Circuits Conf.*, Oct. 2004, pp. 465–468.

[35] C.-C. Wang, C.-J. Cheng, T.-F. Chen, and J.-S. Wang, "An adaptively dividable dual-port BiTCAM for virus-detection processors in mobile devices," *IEEE J. Solid-State Circuits*, vol. 44, no. 5, pp. 1571–1581, May 2009.

[36] C.-C. Wang, J.-S. Wang, and C. Yeh, "High-speed and low-power design techniques for TCAM macros," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 530–540, Feb. 2008.

[37] C. Li et al., "Analog content-addressable memories with memristors," *Nature Commun.*, vol. 11, p. 1638, Apr. 2020, doi: 10.1038/s41467-020-15254-4.

[38] G. Pedretti et al., "Differentiable content addressable memory with memristors," *Adv. Electron. Mater.*, vol. 8, no. 8, Aug. 2022, Art. no. 2101198. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aelm.202101198

[39] T. Li et al., "Design exploration of dynamic multi-level ternary content-addressable memory using nanoelectromechanical relays," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jun. 2023, pp. 1–6.

[40] D. Ielmini and H.-S.-P. Wong, "In-memory computing with resistive switching devices," *Nature Electron.*, vol. 1, no. 6, pp. 333–343, Jun. 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:57248729

[41] B.-D. Yang, Y.-K. Lee, S.-W. Sung, J.-J. Min, J.-M. Oh, and H.-J. Kang, "A low power content addressable memory using low swing search lines," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 12, pp. 2849–2858, Dec. 2011.

[42] P.-T. Huang and W. Hwang, "A 65 nm 0.165 fJ/bit/search 256 × 144 TCAM macro design for IPv6 lookup tables," *IEEE J. Solid-State Circuits*, vol. 46, no. 2, pp. 507–519, Feb. 2011.

[43] H. Noda et al., "A cost-efficient high-performance dynamic TCAM with pipelined hierarchical searching and shift redundancy architecture," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 245–253, Jan. 2005.

[44] N. Gupta, A. Makosiej, H. Shrimali, A. Amara, A. Vladimirescu, and C. Anghel, "Tunnel FET negative-differential-resistance based 1T1C refresh-free-DRAM, 2T1C SRAM and 3T1C CAM," *IEEE Trans. Nanotechnol.*, vol. 20, pp. 270–277, 2021.

[45] Z. Lin et al., "Two-direction in-memory computing based on 10T SRAM with horizontal and vertical decoupled read ports," *IEEE J. Solid-State Circuits*, vol. 56, no. 9, pp. 2832–2844, Sep. 2021.

[46] J. Chen, W. Zhao, Y. Wang, Y. Shu, W. Jiang, and Y. Ha, "A reliable 8T SRAM for high-speed searching and logic-in-memory operations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 30, no. 6, pp. 769–780, Jun. 2022.

[47] A. T. Do, C. Yin, K. S. Yeo, and T. T. Kim, "Design of a power-efficient CAM using automated background checking scheme for small match line swing," in *Proc. ESSCIRC*, Sep. 2013, pp. 209–212.

[48] M. R. Khan and A. H. Rashid, "Memristor-transistor hybrid ternary content addressable memory using ternary memristive memory cell," *IET Circuits, Devices Syst.*, vol. 15, no. 7, pp. 619–629, Oct. 2021.

[49] S. Matsunaga et al., "A 3.14 μm^2 4T-2MTJ-cell fully parallel TCAM based on nonvolatile logic-in-memory architecture," in *Proc. Symp. VLSI Circuits (VLSIC)*, Jun. 2012, pp. 44–45. [Online]. Available: https://api.semanticscholar.org/CorpusID:3449145

[50] X. Yin, K. Ni, D. Reis, S. Datta, M. Niemier, and X. S. Hu, "An ultra-dense 2FeFET TCAM design based on a multi-domain FeFET model," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 9, pp. 1577–1581, Sep. 2019.

[51] J. Bazzi, J. Sweidan, M. E. Fouda, R. Kanj, and A. M. Eltawil, "Efficient analog CAM design," 2022, *arXiv:2203.02500*.

[52] C.-C. Wang, C.-H. Hsu, C.-C. Huang, and J.-H. Wu, "A self-disabled sensing technique for content-addressable memories," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 1, pp. 31–35, Jan. 2010.

[53] H. Noda, K. Inoue, H. J. Mattausch, T. Koide, and K. Arimoto, "A cost-efficient dynamic ternary CAM in 130 nm CMOS technology with planar complementary capacitors and TSR architecture," in *Symp. VLSI Circuits. Dig. Tech. Papers*, 2003, pp. 83–84.

[54] M.-F. Chang, "A 3T1R nonvolatile TCAM using MLC ReRAM with sub-1ns search time," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2015, pp. 1–3.

[55] J. Li, R. K. Montoye, M. Ishii, and L. Chang, "1 mb 0.41 μm^2 2T-2R cell nonvolatile TCAM with two-bit encoding and clocked self-referenced sensing," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 896–907, Apr. 2014.

[56] C.-C. Lin et al., "A 256b-wordlength ReRAM-based TCAM with 1ns search-time and 14× improvement in wordlength-energyefficiency-density product using 2.5T1R cell," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 136–137.

[57] L.-Y. Huang et al., "ReRAM-based 4T2R nonvolatile TCAM with 7× NVM-stress reduction, and 4× improvement in speed-wordlength-capacity for normally-off instant-on filter-based search engines used in big-data processing," in *Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2014, pp. 1–2.

[58] M. Chang et al., "A 3T1R nonvolatile TCAM using MLC ReRAM for frequent-off instant-on filters in IoT and big-data processing," *IEEE J. Solid-State Circuits*, vol. 52, no. 6, pp. 1664–1679, Jun. 2017.

[59] B. Song, T. Na, J. P. Kim, S. H. Kang, and S.-O. Jung, "A 10T-4MTJ nonvolatile ternary CAM cell for reliable search operation and a compact area," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 6, pp. 700–704, Jun. 2017.

[60] S. Matsunaga et al., "Design and fabrication of a one-transistor/one-resistor nonvolatile binary content-addressable memory using perpendicular magnetic tunnel junction devices with a fine-grained power-gating scheme," *Jpn. J. Appl. Phys.*, vol. 50, no. 6R, Jun. 2011, Art. no. 063004.

[61] S. Matsunaga, A. Katsumata, M. Natsui, T. Endoh, H. Ohno, and T. Hanyu, "Design of a nine-transistor/two-magnetic-tunnel-junction-cell-based low-energy nonvolatile ternary content-addressable memory," *Jpn. J. Appl. Phys.*, vol. 51, no. 2S, 2012, Art. no. 02BM06.

[62] W. Xu, T. Zhang, and Y. Chen, "Design of spin-torque transfer magnetoresistive RAM and CAM/TCAM with high sensing and search speed," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 1, pp. 66–74, Jan. 2010.

[63] S. Matsunaga et al., "Fully parallel 6T-2MTJ nonvolatile TCAM with single-transistor-based self match-line discharge control," in *IEEE Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2011, pp. 298–299.

[64] L. Xue, Y. Cheng, J. Yang, P. Wang, and Y. Xie, "ODESY: A novel 3T-3MTJ cell design with optimized area density, scalability and latency," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2016, pp. 1–8.

[65] X. Yin, M. Niemier, and X. S. Hu, "Design and benchmarking of ferroelectric FET based TCAM," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 1444–1449.

[66] X. Yin et al., "FeCAM: A universal compact digital and analog content addressable memory using ferroelectric," *IEEE Trans. Electron Devices*, vol. 67, no. 7, pp. 2785–2792, Jul. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:214802295

[67] K. Ni et al., "Ferroelectric ternary content-addressable memory for one-shot learning," *Nat. Electron.*, vol. 2, no. 11, pp. 521–529, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:257095008

[68] S. Veeramani, M. Kumar, and S. K. N. Mahammad, "Hybrid trie based partitioning of TCAM based openflow switches," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2013, pp. 1–5.

[69] M. Moradi, F. Qian, Q. Xu, Z. M. Mao, D. Bethea, and M. K. Reiter, "Caesar: High-speed and memory-efficient forwarding engine for future internet architecture," in *Proc. ACM/IEEE Symp. Architectures Netw. Commun. Syst. (ANCS)*, May 2015, pp. 171–182.

[70] M. Imani, Y. Kim, A. Rahimi, and T. Rosing, "ACAM: Approximate computing based on adaptive associative memory with online learning," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2016, pp. 162–167.

[71] M. Imani, D. Peroni, Y. Kim, A. Rahimi, and T. Rosing, "Efficient neural network acceleration on GPGPU using content addressable memory," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2017, pp. 1026–1031.

[72] W. Choi, K. Jeong, K. Choi, K. Lee, and J. Park, "Content addressable memory based binarized neural network accelerator using time-domain signal processing," in *Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6.

[73] C. E. Graves et al., "In-memory computing with memristor content addressable memories for pattern matching," *Adv. Mater.*, vol. 32, no. 37, Sep. 2020, Art. no. 2003437.

[74] S. Park, J.-W. Nam, and S. K. Gupta, "HW-BCP: A custom hardware accelerator for sat suitable for single chip implementation for large benchmarks," in *Proc. 26th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, 2021, pp. 29–34.

[75] X. Yin et al., "Deep random forest with ferroelectric analog content addressable memory," 2021, *arXiv:2110.02495*.

[76] L. Chisvin and R. J. Duckworth, "Content-addressable and associative memory: Alternatives to the ubiquitous RAM," *Computer*, vol. 22, no. 7, pp. 51–64, Jul. 1989.

[77] L. S. Kida, "Associative processing implemented with content-addressable memories," Master's thesis, Dept. Elect. Eng., Portland State Univ., Jul. 1991, doi: 10.15760/etd.6060.

[78] L. H. B. Sardinha, D. S. Silva, M. A. M. Vieira, L. F. M. Vieira, and O. P. Vilela Neto, "TCAM/CAM-QCA: (Ternary) content addressable memory using quantum-dot cellular automata," *Microelectron. J.*, vol. 46, no. 7, pp. 563–571, Jul. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0026269215000798

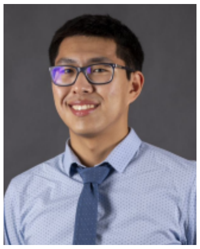
[79] S. R. Heikalabad, A. H. Navin, and M. Hosseinzadeh, "Content addressable memory cell in quantum-dot cellular automata," *Microelectron. Eng.*, vol. 163, pp. 140–150, Sep. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016793171630332X

[80] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 305–316, Sep. 2005.

[81] S. Pontarelli, M. Ottavi, and A. Salsano, "Error detection and correction in content addressable memories by using Bloom filters," in *Proc. IEEE 25th Int. Symp. Defect Fault Tolerance VLSI Syst.*, vol. 62, Nov. 2010, pp. 420–428.

[82] A. Varada and S. Agrawal, "An efficient SRAM-based ternary content addressable memory (TCAM) with soft error correction," in *Proc. 5th Int. Conf. Electron., Mater. Eng. Nano-Technol. (IEMENTech)*, Sep. 2021, pp. 1–6.

[83] R. M. Roth, "Error-detection schemes for analog content-addressable memories," *IEEE Trans. Comput.*, vol. 73, no. 7, pp. 1795–1808, Jul. 2024.



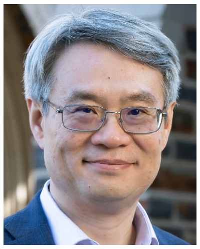
Tergel Molom-Ochir received the B.S. degree in electrical engineering from the University of Massachusetts Amherst, MA, USA, in 2023. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA, under the supervision of Prof. Yiran Chen. His research interests include circuit design, in-memory computing, AI accelerators, and non-volatile memories.

Brady Taylor (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from Rice University in 2019 and the M.S. degree in electrical and computer engineering from Duke University in 2022, where he is currently pursuing the Ph.D. degree under the guidance of Dr. Helen Li. His research interests include mixed-signal circuit design, computer architecture, neuromorphic computing, and emerging nanoelectronic devices.

Hai (Helen) Li (Fellow, IEEE) received the Ph.D. degree from Purdue University in 2004. She is currently the Marie Foote Reel E'46 Distinguished Professor and the Department Chair of the Electrical and Computer Engineering Department, Duke University.

Her current research interests include neuromorphic circuits and systems for brain-inspired computing, machine learning acceleration and trustworthy AI, conventional and emerging memory design and architecture, and software and hardware co-design.

She was a recipient of the NSF Career Award in 2012, the DARPA Young Faculty Award in 2013, the TUM-IAS Hans Fischer Fellowship from Germany in 2017, and the ELATE Fellowship in 2020. She received nine best paper awards and additional nine best paper nominations from international conferences. She was a Distinguished Lecturer of the IEEE CAS Society from 2018 to 2019 and a Distinguished Speaker of ACM from 2017 to 2020.



Yiran Chen (Fellow, IEEE) received the Ph.D. degree from Purdue University in 2005. He is currently the John Cocke Distinguished Professor of electrical and computer engineering at Duke University and the Director of the NSF AI Institute for Edge Computing Leveraging the Next-Generation Networks (Athena), the NSF Industry-University Cooperative Research Center (IUCRC) for Alternative Sustainable and Intelligent Computing (ASIC), and the Co-Director of Duke Center for Computational Evolutionary Intelligence (DCEI).

He received 11 best paper awards, one best poster award, and 15 best paper nominations from international conferences and workshops.

He received numerous awards for his technical contributions and professional services, such as the IEEE CASS Charles A. Desoer Technical Achievement Award and the IEEE Computer Society Edward J. McCluskey Technical Achievement Award. He has been the Distinguished Lecturer of IEEE CEDA and CASS, and the Distinguished Visitor of IEEE CS.